N90-29877

# KINEMATICS & CONTROL ALGORITHM DEVELOPMENT AND SIMULATION FOR A REDUNDANT TWO-ARM ROBOTIC MANIPULATOR SYSTEM

Michael P. Hennessey, Paul C. Huang, and Charles T. Bunnell
Advanced Systems Center
FMC Corporation
1200 South Second Street
Minneapolis, MN 55459-0043

**Abstract**

An efficient approach to cartesian motion and force control of a 7 degree of freedom (dof) manipulator is presented. It is based on extending the active stiffness controller to the 7 dof case in general and use of an efficient version of the gradient projection technique for solving the inverse kinematics problem. Cooperative control is achieved through appropriate configuration of individual manipulator controllers. In addition, other aspects of trajectory generation using standard techniques are integrated into the controller. The method is then applied to a specific manipulator of interest (Robotics Research T-710). Simulation of the kinematics, dynamics, and control are provided in the context of several scenarios; one pertaining to a noncontact pick and place operation, one relating to contour following where contact is made between the manipulator and the environment, and one pertaining to cooperative control.

## 1 Introduction

Cartesian motion and force control of a manipulator is needed for many different types of automation applications such as material handling and assembly [1]. Because of the complexity of some potential applications (e.g. in space and in certain military environments) and because of the inherent limitations of many 6 dof manipulators (e.g. singularity problems), 7 (or more) dof manipulators are being proposed for these applications. As a result, there are many interesting and challenging problems, particularly with respect to kinematics, control algorithms, and controller implementation aspects. Kinematic problems stem largely from two sources: (1) the inverse positional kinematics solution is not unique, and (2) it typically does not exist in closed form. As a consequence of the nonclosed form issue, control is complicated from the standpoint that highly modular approaches may not be viable (e.g. use of individual joint position servos). Also, despite the continual increase in performance and decrease in cost of controller hardware, algorithm efficiency is still an issue. Below, the focus is on efficient motion and force control of a 7 dof articulated manipulator.

For the 7 dof manipulator case, few kinematic configurations permit closed form inverse positional (in general "location") kinematic solutions. For the remaining cases, other approaches must be taken, such as use numerical iteration to solve for the inverse positional kinematic solution or knowing the inverse Jacobian, servo at the cartesian level. Baker and Wampler [2] refer to all kinematic inversion methods as either "global" or "local" methods. In both cases, the redundancy can be used to optimize a criterion of interest. With respect to the first approach, convergence and computational efficiency can be a serious issue and perhaps somewhat suprisingly, it may not always be necessary to calculate the joint angles corresponding to a particular end effector location, hence obviating the need to solve the inverse positional kinematics problem. The second approach avoids the difficulties associated with inverting the positional kinematics, but requires a different controller implementation. A recent technique in this catagory is the gradient projection technique [2,3].

Many different control algorithms have been proposed for motion and force control of a manipulator, including: (1) hybrid control, (2) modified hybrid control, (3) active stiffness control, (4) modified resolved acceleration control, (5) impedance control, (6) operational space control, (7) free joint control, and (8) modified free joint control [4]. At first glance it appears as though there are many radically different control schemes, when in fact there are not [5]. All of the above control laws fit into roughly only two catagories: (1) "hybrid" control and (2) active stiffness control. However, at a practical level both types of controllers can perform hybrid type control depending on how the control system is

configured. In hybrid control, position and force are controlled in basically orthogonal directions, and in stiffness control the nominal position is controlled and its endpoint stiffness specified.

Based on the above discussion there are many different approaches for solving the 7 dof manipulator kinematics and control problem. In the interests of stability coupled with a desire to minimize computational requirements (for implementation reasons primarily), the basic approach taken in this paper is to combine the active stiffness controller with an efficient version of the gradient projection technique. Also, since its origination, the active stiffness controller (or a slight variation of it) has been experimentally verified to work on several nonredundant manipulators and is well known for being computationally efficient (e.g. in [6]). By appropriately configuring individual manipulator controllers, cooperative control can be achieved.

# 2 General Formulation

The general block diagram of the system showing the active stiffness controller extended to 7 dof is depicted in Figure 1. The controller consists of the following major elements: (1) location feedback loop, (2) joint rate feedback loop, (3) force feedback loop, (4) torque compensation, (5) calculation of location error, (6) direct kinematics, (7) gravity compensation, and (8) trajectory generator. The location (position and orientation) feedback loop determines the manipulator's effective stiffness and differs from the original formulation in that the servoing is done at the cartesian level because of the nonclosed form inverse positional kinematics issue. To determine the cartesian feedback contribution we observe that the location feedback given by [6] is:

$$K_\theta \delta\underline{\theta} \equiv (J^T K J)\delta\underline{\theta} \equiv (J^T K)(J\delta\underline{\theta}) \equiv (J^T K)\delta\underline{X} \tag{1}$$

so that $J^T K$ is the cartesian location feedback. Here $J$ is the manipulator Jacobian of the end effector location expressed in world coordinates, $K$ is the diagonal arm stiffness matrix, $\delta\theta$ is the joint error vector, and $\delta X$ is the resulting cartesian error vector expressed in world coordinates. We note that the feedback from Equation (1) ignores the nullspace of the Jacobian. The joint rate feedback loop (needed for stability purposes), the force feedback loop, and the torque compensation loop characterized by $G$ which is especially useful for contact situations, follow essentially as before [6]. Calculation of the location error is defined by $\Delta$, the 4x4 homogeneous error transformation given by:

$$\Delta = T_c T_a^{-1} = \begin{bmatrix} 1 & -\delta z & \delta y & \Delta x \\ \delta z & 1 & -\delta x & \Delta y \\ -\delta y & \delta x & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where $T_c$ and $T_a$ are the commanded and actual manipulator transformations and $\delta\underline{X}^T = [\Delta x, \Delta y, \Delta z, \delta x, \delta y, \delta z]$. We note that $\Delta$ must be in world coordinates and that average values for $\delta x, \delta y$, and $\delta z$ may be derived as implied by Equation (2). Finally, the direct kinematics, gravity compensation, and the trajectory generator will be developed in conjunction with application to a particular manipulator.

# 3 Application to a Particular 7 DOF Manipulator

The general formulation will now be applied to the Robotics Research T-710 manipulator [7].

## 3.1 Direct Kinematics

The location of the 7th joint's coordinate frame in base coordinates ( $T^7_{base}$) may be expressed as the product of successive 4x4 homogeneous transformation matrices ($A^i_{i-1}$) and is given by [8,9]:

$$T^7_{base} = A^1_{base} A^2_1 A^3_2 A^4_3 A^5_4 A^6_5 A^7_6 \tag{3}$$

where:

$$A^i_{i-1} = \begin{bmatrix} P_i & \underline{r}_i \\ \underline{0} & 1 \end{bmatrix} \tag{4}$$

with $P_i$ and $\underline{r}_i$ given recursively by:

$$P_i = P_{i-1} P_{i-1,i} = P_{i-1} P^{Ref}_{i-1,i} P^{\theta_i}_{i-1,i} = P_{i-1} P^{Ref}_{i-1,i} \begin{bmatrix} c_i & 0 & s_i \\ 0 & 1 & 0 \\ -s_i & 0 & c_i \end{bmatrix} \tag{5}$$

and:

$$\underline{r}_i = \underline{r}_{i-1} + P_{i-1} \underline{h}_{i-1,i} \tag{6}$$

Here $c_i \equiv \cos\theta_i, s_i \equiv \sin\theta_i$, $\underline{h}_{i-1,i}$ is the vector from the origin of the $i-1th$ coordinate frame to the $ith$ coordinate frame expressed in the $i-1th$ coordinate frame, and $\underline{r}_i$ denotes the position vector of joint $i$ in world coordinates with

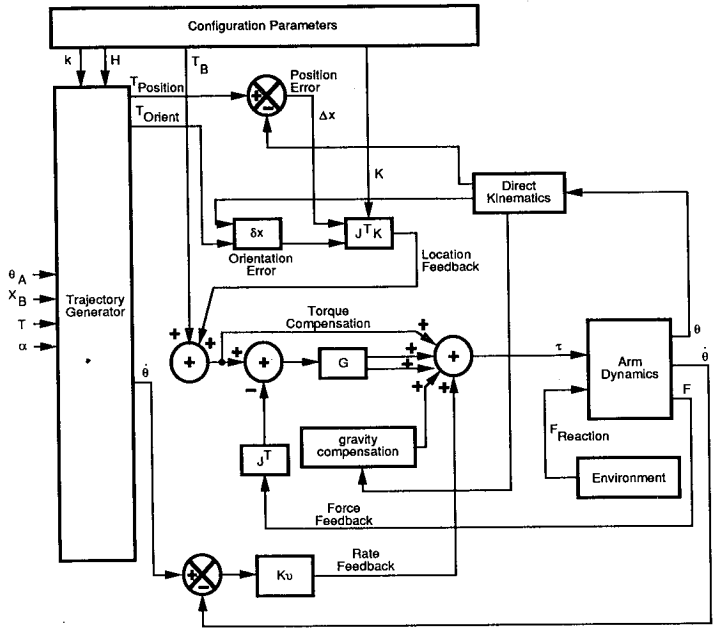



Figure 1 Block diagram of manipulator controller and system

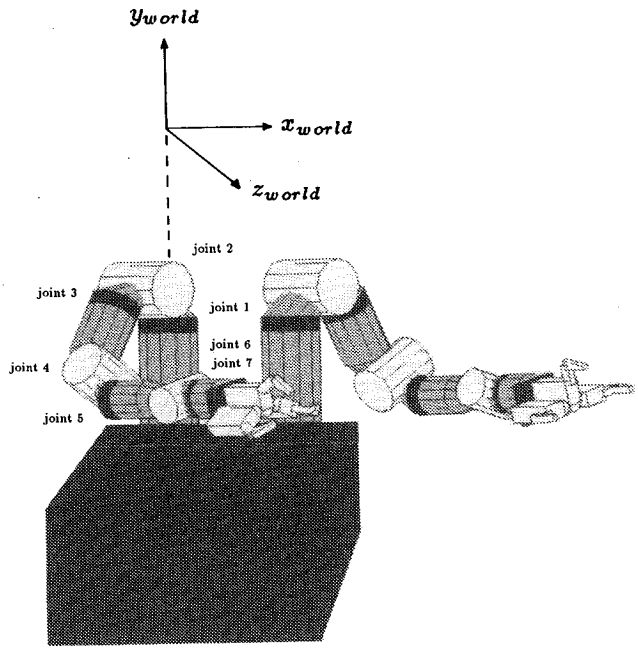



Figure 2(a) Kinematic configuration of several Robotic Research T-710 manipulators

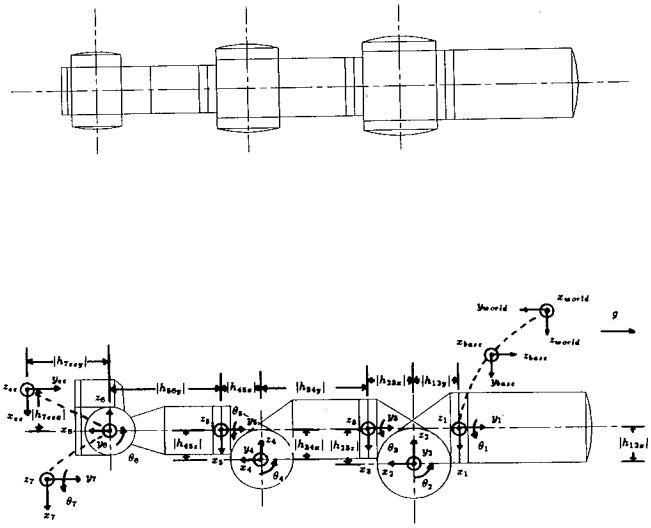
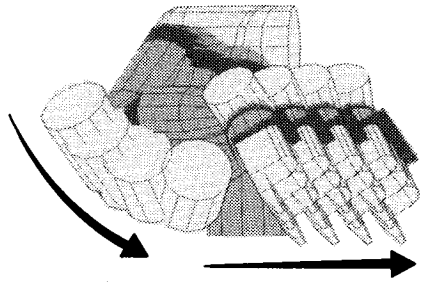
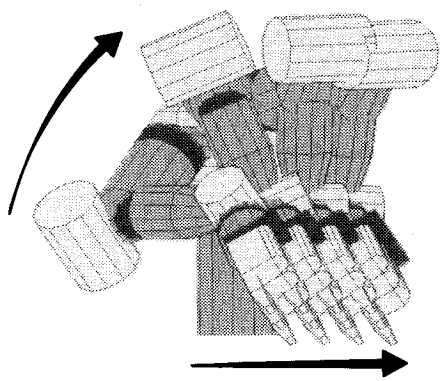


Figure 2(b) Direct kinematics of Robotic Research T-710 manipulator



(a) straight line motion without "elbow-up" potential function



(b) straight line motion with "elbow-up" potential function

Figure 3 Illustration of gradient projection technique using "elbow-up" potential function

representative kinematic data (i.e. $P^{Ref}_{i-1,i}, \underline{h}_{i-1,i}$) given in the Appendix. For the Robotics Research T-710 manipulator shown schematically in Figure 2 the total manipulator transformation $T^7_{base}$ may be shown to be given by (excluding the base and end effector transformations):

$$T^7_{base} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

where the positional entries are given by:

$$\begin{aligned} p_x &= -(c_1 s_3 + s_1 c_2 c_3)(h_{34x} + h_{45x} s_4 - h_{45z} c_4 - h_{56y} s_4) \\ &\quad + s_1 s_2 (h_{34y} - h_{45x} c_4 - h_{45z} s_4 + h_{56y} c_4) \\ &\quad - h_{23x} s_1 s_2 + h_{23z} s_1 c_2 - h_{12x} s_1 \end{aligned}$$

$$\begin{aligned} p_y &= (-s_1 s_3 + c_1 c_2 c_3)(h_{34x} + h_{45x} s_4 - h_{45z} c_4 - h_{56y} s_4) \\ &\quad - c_1 s_2 (h_{34y} - h_{45x} c_4 - h_{45z} s_4 + h_{56y} c_4) \\ &\quad + h_{23x} c_1 s_2 - h_{23z} c_1 c_2 + h_{12x} c_1 \end{aligned} \tag{8}$$

$$\begin{aligned} p_z &= s_2 c_3 (h_{34x} + h_{45x} s_4 - h_{45z} c_4 - h_{56y} s_4) \\ &\quad + c_2 (h_{34y} - h_{45x} c_4 - h_{45z} s_4 + h_{56y} c_4) \\ &\quad - h_{23x} c_2 - h_{23z} s_2 + h_{12y} \end{aligned}$$

We note that Equation (8) represents three equations in four unknowns or the mechanical decoupling of the major and minor portion of the manipulator linkage. While not proven here, because of certain offsets (i.e. $\underline{h}_{i-1,i}$) these equations can not be solved in closed form. With $T^7_{base}$ defined from above we can include the base and end effector transformations. For now, let:

$$A^{base}_{world} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

and:

$$A^{ee}_7 = \begin{bmatrix} 1 & 0 & 0 & h_{7eex} \\ 0 & 1 & 0 & h_{7eey} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

so that the entire transformation ($T$) is given by:

$$T = A^{base}_{world} T^7_{base} A^{ee}_7 \tag{11}$$

## 3.2 Gravity Compensation

Gravity compensation is used to null out the effects of gravity loading of the manipulator, thereby lessening the role of feedback and improving performance. Of course for space applications this is not needed. Compensation is achieved by calculating anticipated torques driven by gravity only and uses the recursive Newton-Euler formulation. These torques ($\underline{\tau}_{gravity}$) are given by [9]:

$$\tau_{gravity_i} = \underline{s}_i^T \underline{t}_i \tag{12}$$

where:

$$\underline{t}_i = \underline{t}_{i+1} + (\underline{r}_{i+1} - \underline{r}_i) \times \underline{f}_{i+1} + \underline{h}_{i,i} \times m_i \underline{g} \tag{13}$$

$$\underline{f}_i = \underline{f}_{i+1} + m_i \underline{g} \tag{14}$$

$$\underline{h}_{i,i} = P_i \underline{r}_{cg_{i,i}} \tag{15}$$

Here $\underline{s}_i$ is the *ith* joint axis unit vector in world coordinates, $\underline{r}_{cg_{i,i}}$ is the center of gravity vector of the *ith* link in the *ith* coordinate frame, $\underline{f}_i$ and $\underline{t}_i$ are the applied forces and torques to joint $i$, $m_i$ is the mass of link $i$, and $\underline{g}$ is the gravity vector expressed in world coordinates ($[0, -9.81 m/sec^2, 0]^T$).

## 3.3 Trajectory Generator

The purpose of the trajectory generator is to provide the controller with smooth input trajectories. In general this applies to both cartesian (in world or end effector coordinates) and joint interpolated motion for both autonomous and teleoperated cases. The focus here is on the most difficult and interesting case; namely autonomous cartesian control (in world coordinates). Figure 1 illustrates the specific inputs and outputs of the trajectory generator. Inputs are the initial joint angles ($\theta_A$) (at point "A"), the commanded cartesian location transformation ($T_B$) (point "B" expressed as three positions and three Euler angles), the travel time (T), and a normalized acceleration parameter ($\alpha \geq 4/\mathrm{T}$) - assume a symmetric trapezoidal velocity profile. Time varying outputs are the location transformation ($T(t)$) and the joint rates ($\dot{\theta}(t)$). The location transformation may be derived without knowledge of the inverse kinematics and $\dot{\theta}(t)$ will be derived using the gradient projection technique. Also, when properly configured the trajectory generator can be used to solve the inverse positional kinematics problem by providing an initial guess for the joint angles and saving the last set of angles available internally to the trajectory generator. Of course the solution is not unique.

### 3.3.1 Location Transformation

To permit smooth simultaneous changes in orientation and position of the end effector, the required orientation change takes place about a constant vector $\underline{k}$ in world coordinates while the origin of the end effector's local coordinate frame is translating along a straight line. In both cases, trapezoidal velocity profiles are used to smooth the motion. This is equivalent to use of the following transformation to describe the end effector's time varying location:

$$T(t) = \begin{bmatrix} Rot(\underline{k}, \theta_k(t)) & \underline{p}(t) \\ \underline{0} & 1 \end{bmatrix} \tag{16}$$

where $\dot{\theta}_k, \dot{\underline{p}} = [\dot{x}, \dot{y}, \dot{z}]^T$ have trapazoidal velocity profiles (for $0 \leq t \leq$T) with symmetric ramp slopes $\pm\alpha$ and $Rot(\underline{k}, \theta_k)$ is a rotation matrix corresponding to a rotation of $\theta_k$ about a constant vector $\underline{k}$ in world coordinates given by [8]:

$$Rot(\underline{k}, \theta_k) = \begin{bmatrix} k_x k_x vers\theta_k + \cos\theta_k & k_y k_x vers\theta_k - k_z \sin\theta_k & k_z k_x vers\theta_k + k_y \sin\theta_k \\ k_x k_y vers\theta_k + k_z \sin\theta_k & k_y k_y vers\theta_k + \cos\theta_k & k_z k_y vers\theta_k - k_x \sin\theta_k \\ k_x k_z vers\theta_k - k_y \sin\theta_k & k_y k_z vers\theta_k + k_x \sin\theta_k & k_z k_z vers\theta_k + \cos\theta_k \end{bmatrix} \tag{17}$$

Here $vers\theta_k = 1 - \cos\theta_k$. The vector $\underline{k} = [k_x, k_y, k_z]^T$ and $\theta_k$(T) are evaluated in detail in Paul [8] and use the following net rotation matrix as input:

$$\Delta_{orientation} = T_{B3x3} T_{A3x3}^{-1} \tag{18}$$

with $T_{A3x3}, T_{B3x3}$ referring to their respective orientation submatrices.

### 3.3.2 Efficient Gradient Projection Technique

**Overview** In addition to using the manipulator's joints to permit cartesian end effector motion the manipulator because of its redundancy possesses self-motion; that is, the joints can move without the end effector moving. The gradient projection technique uses this motion of the manipulator to attempt to optimize a criterion of interest and has been proposed recently by several researchers [2,3]. The self-motion of the arm is characterized by the nullspace of the Jacobian (i.e. $\dot{\underline{\theta}} | J\dot{\underline{\theta}} = \underline{0}$) and it is desired to express the self-motion joint rates as the *gradient* of a potential function $H$ of interest *projected* onto the nullspace so that:

$$\dot{\underline{\theta}}_{null} = (I - J^{\dagger}J) k \nabla H \tag{19}$$

where $J^{\dagger}$ is the Moore-Penrose psuedo-inverse of $J$ (i.e. $J^T[JJ^T]^{-1}$) and $k$ is the nullspace gain. In the interests of implementation, an optimized version of the gradient projection technique is applied to the manipulator (i.e. $J^{\dagger}$ is never actually computed). It was developed and is described in detail in Dubey et al [3]. Basically, the technique takes advantage of the existence of a 3x3 $\underline{0}$ block in the Jacobian for spherical wristed manipulators and an assumption regarding singularities induced by the remaining 4 joints. Specifically, which of the first four columns of the Jacobian when removed still permits the remaining matrix to be invertible. The end result consists of several simplified sets of 3 linear equations in 3 unknowns to be solved, which can be done quite easily.[1]

[1] A practical detail concerns the use of a numerical integrator to estimate $\underline{\theta}(t)$ for use in evaluating the inverse Jacobian. As an aside, this estimate is subject to drift and is not accurate enough to serve as a joint position command signal, but good enough for evaluating $J$. With respect to integration techniques, both a fourth order Runge-Kutta and a simple Euler integrator have been used with success for reasonable speeds (for actual implementation the Euler integrator is perferred because of its simplicity).

**Application to Robotics Research T-710 Manipulator** Major inputs to the efficient gradient projection technique are the manipulator Jacobian which for efficiency reasons only is expressed in the third coordinate frame for the wrist (not the end effector) and the desired potential function. We will also have reason to evaluate the world to third coordinate frame transformation, the end effector to wrist velocity transformation and for convenience $J$ will be evaluated in this section (recall that it is needed in the location and force feedback loop).

The $ith$ $(i = 1, 2, ..., 7)$ column of the wrist Jacobian expressed in the third coordinate frame is given by:

$$\left[ \; J_3^w \; \right]_i = \begin{bmatrix} \underline{s}_{3i} \times (\underline{r}_{3w} - \underline{r}_{3i}) \\ \underline{s}_{3i} \end{bmatrix} \tag{20}$$

where $\underline{s}_{3i}, \underline{r}_{3i}$, and $\underline{r}_{3w}$ are the $ith$ joint axis unit vector, the ith joint location vector, and the wrist location vector expressed in the third coordinate frame, respectively. For the Robotics Research T-710 manipulator the Jacobian $J_3^w$ is given by:

$$J_3^w = \begin{bmatrix} (J_3^w)_{11} & (J_3^w)_{12} & 0 & h_{34x}s_4 + h_{45x}c_4 - h_{56y}c_4 & 0 & 0 & 0 \\ (J_3^w)_{21} & (J_3^w)_{22} & 0 & -h_{34x}c_4 + h_{45x}s_4 - h_{56y}s_4 & 0 & 0 & 0 \\ (J_3^w)_{31} & (J_3^w)_{32} & (J_3^w)_{33} & 0 & 0 & 0 & 0 \\ s_2c_3 & -s_3 & 0 & 0 & -s_4 & c_4s_5 & -s_4c_6 - c_4c_5s_6 \\ c_2 & 0 & 1 & 0 & c_4 & s_4s_5 & c_4c_6 - s_4c_5s_6 \\ s_2s_3 & c_3 & 0 & 1 & 0 & c_5 & s_5s_6 \end{bmatrix} \tag{21}$$

where the lengthy entries are given in [10].

Next, an example use of the potential function will be given. In general one may envision it to be some combination (likely a weighted sum) of various subpotential functions associated with singularity avoidance, joint limit avoidance, or other criteria of interest of a heuristic nature. To attempt to avoid the wrist singularity, one could choose a potential function like $H = s_6^2$ to be maximized. For the joint limit problem, one may attempt to minimize $H = \sum_{i=1}^{7}(\theta_i - \theta_{icenter})^2$. Finally, as a heuristic example, if it is desired to keep the elbow "up" perhaps for collision avoidance reasons, one would like the $y_{world}$ component of the $-y_3$ axis (i.e. $c_2$) to be large (see Figure 2), so choose $H = (c_2 + 1)^2$ to be maximized with $k = 1$. Figure 3 shows the effect of this particular potential function as the manipulator's end effector traverses along a straight line.

The transformation projecting the end effector velocities in the world coordinate frame ($\dot{\underline{X}}$) onto the third coordinate frame is given by:

$$\dot{\underline{X}}_3^{ee} \equiv \begin{bmatrix} \underline{V}_3^{ee} \\ \underline{\omega}_3^{ee} \end{bmatrix} = \begin{bmatrix} R_{world}^3 & \underline{0} \\ \underline{0} & R_{world}^3 \end{bmatrix} \dot{\underline{X}} \equiv \begin{bmatrix} R_{world}^3 & \underline{0} \\ \underline{0} & R_{world}^3 \end{bmatrix} \begin{bmatrix} \underline{V} \\ \underline{\omega} \end{bmatrix} \tag{22}$$

where:

$$R_{world}^3 = \begin{bmatrix} -c_1s_3 - s_1c_2c_3 & -s_2c_3 & -s_1s_3 + c_1c_2c_3 \\ s_1s_2 & -c_2 & -c_1s_2 \\ c_1c_3 - s_1c_2s_3 & -s_2s_3 & s_1c_3 + c_1c_2s_3 \end{bmatrix} \tag{23}$$

The end effector to wrist velocity transformation allows one to transform commanded end effector velocities into corresponding wrist velocities and is given by:

$$\underline{V}_3^w = \underline{V}_3^{ee} - \underline{\omega}_3^{ee} \times \underline{r}_3^{ee} \tag{24}$$

$$\underline{\omega}_3^w = \underline{\omega}_3^{ee} \tag{25}$$

where $\underline{r}_3^{ee}$ is given by:

$$\underline{r}_3^{ee} = \begin{bmatrix} h_{7eex}[-c_4s_5s_7 - c_7(s_4s_6 - c_4c_5c_6)] - h_{7eey}(s_4c_6 + c_4c_5s_6) \\ h_{7eex}[-s_4s_5s_7 + c_7(c_4s_6 + s_4c_5c_6)] + h_{7eey}(c_4c_6 - s_4c_5s_6) \\ -h_{7eex}(c_5s_7 + s_5c_6c_7) + h_{7eey}s_5s_6 \end{bmatrix} \tag{26}$$

and $\underline{V}_3^w$ and $\underline{\omega}_3^w$ are the translational and rotational velocities of the wrist expressed in the third coordinate frame.

Having already evaluated the Jacobian needed for trajectory generation $(J_3^w)$, by using the above relationships we can evaluate the Jacobian needed for location and force feedback $(J)$ which describes the end effector cartesian rates in the world coordinate frame by means of:

$$J = \begin{bmatrix} (R_{world}^3)^T(J_3^w)_V + (R_{world}^3)^T \left[ \; (J_3^w)_{\omega 1} \times \underline{r}_3^{ee} \quad (J_3^w)_{\omega 2} \times \underline{r}_3^{ee} \quad \cdots \quad (J_3^w)_{\omega 7} \times \underline{r}_3^{ee} \; \right] \\ (R_{world}^3)^T(J_3^w)_\omega \end{bmatrix} \tag{27}$$

where:

$$J_3^w \equiv \begin{bmatrix} (J_3^w)_{V(3x7)} \\ (J_3^w)_{\omega(3x7)} \end{bmatrix} \equiv \begin{bmatrix} (J_3^w)_{V(3x7)} \\ \left[ \; (J_3^w)_{\omega 1(3x1)} \quad (J_3^w)_{\omega 2(3x1)} \quad \cdots \quad (J_3^w)_{\omega 7(3x1)} \; \right] \end{bmatrix} \tag{28}$$

# 4 Kinematics, Dynamics, and Control (KDAC) Simulation

To determine anticipated performance of the above described controller and gain confidence in the approach, various simulation tools were developed/aquired and the kinematics, dynamics, and control were simulated for the Robotics Research T-710 manipulator. The simulation tools of KDAC consist primarily of MATRIXx/AR (Automation and Robotics) [9] augmented with application "blocks" for performing kinematic and control functions and a graphics animation package for displaying the results. Rigid body manipulator dynamics are modeled using appropriately configured blocks arranged to support the recursive Newton-Euler formulation of dynamics. Ideal actuators are assumed as an initial baseline, although this need not be a constraint in the simulation (e.g. actuator inertia and frictional effects can be easily included). Representative manipulator data used in the simulation are given in the Appendix. Several scenarios were simulated based on Jackson [11] which are representative of tasks required of general purpose 7 dof articulated manipulators. The first scenario is concerned with noncontact cartesian position control, the second pertains to contour following of a flat, compliant surface, and the third pertains to cooperative control. While a formal stability analysis was performed only for the first scenario, for the other scenarios, the effects of such parameters as the controller frequency, the stiffness matrix, and the environmental stiffness were investigated empirically.

## 4.1 Cartesian Position Control

Figure 4 illustrates the performance of the system (position command/actual plots) as the end effector attempts to move along a straight line approximately in the $x_{world}$ direction while retaining its orientation with an average speed of 6 inches/sec. The magnitude of the worst tracking error is 4 mm and after an additional 0.1 second the error drops to less than 1 mm (for all directions). When the controller frequency is changed from 100 Hz to 25 Hz the behavior of the system is virtually identical (not shown here). Also, when the stiffness matrix is decreased uniformly by a factor of 100, one notices a fairly insignificant degradation in performance as shown in Figure 5. In conclusion, for a modest speed of 6 inches/sec, the system is fairly robust to changes in the controller frequency and the stiffness matrix when performing noncontact straight line motion. A local stability analysis was performed using a continous linear model at a particular location. The resultant eigenvalues of the closed loop system were all stable (i.e. in left half plane) and when the rate gain was set to zero, all of the eigenvalues were on the imaginary axis. As expected then the rate gain introduces damping and pulls the open loop poles off of the $j\omega$ axis into the left half plane.

## 4.2 Contour Following

Contour following is a generic process in which the manipulator's end effector maintains contact with the surface of the environment while traversing along it. For the specific simulations performed, the manipulator attempted to maintain contact with a flat compliant surface parallel to the world xz plane while moving laterally along the surface approximately in the $x_{world}$ direction. Modeling of the interaction between the manipulator and the environment assumed a massless spring; in general an upgrade of this would be to include mass and damping effects (i.e. "impedance"). Figure 6 shows the nominal performance of the system for an environmental stiffness ($K_{env}$) of 10 kN/m. It is that of a stable lightly damped system (some oscillation). As the controller frequency decreases to 25 Hz the performance degrades considerably (see Figure 7) and is unstable. For a very stiff environment (i.e. $K_{env} \geq 1MN/m$) the system is asymptotically stable (Figure 8), but with considerable oscillation. Both the controller frequency and the environmental stiffness are seen to significantly affect the performance of the system.

## 4.3 Cooperative Control

Modeling cooperative was treated as an extension of the manner in which contour following was modeled. To simplify matters somewhat (i.e. ignore certain coupling effects), the interaction between the two manipulators was modeled as a massless spring and only translational motion was considered. Figure 9 shows the performance of the system as both manipulators laterally translate 0.94 m in the $-x_{world}$ direction while attempting to maintain a 10 $N$ compression force on the spring (in the $x_{world}$ axis direction) that ties the two arms together. Wihin numerical limits, convergence to the steady-state force level is achieved after 0.6 sec of the 3 second trajectory.

# 5 Implementation Issues

Of primary importance in the interests of actual implementation are the details of the bridge between the simulation environment and actual implementation and the computational burden of the above algorithms. This is especially true since for convenience purposes a hybrid approach was taken in developing the necessary simulation blocks (i.e. some are standard blocks and some are custom blocks). Of course, there are many other implementation issues, such as hardware considerations, etc. which will not be discussed here. With respect to the first issue, for implementation purposes only the gravity compensation and the direct kinematics block would follow the algorithm based on the standard blocks

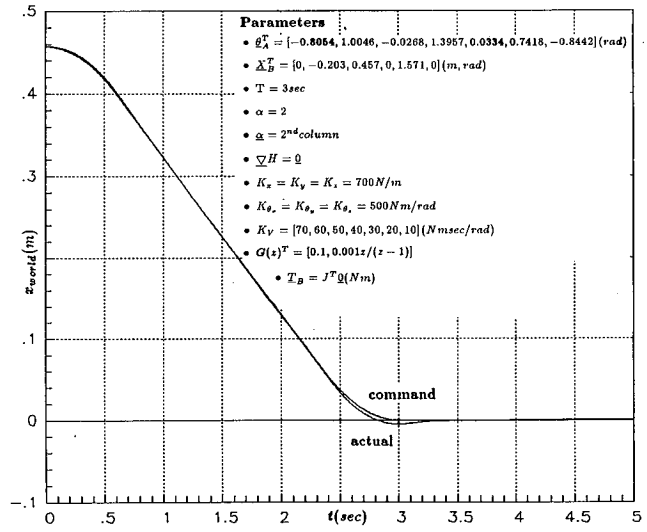



**Figure 4** Position ($x_{world}$) command/actual versus time – scenario number 1 (100 Hz, nominal $K$)

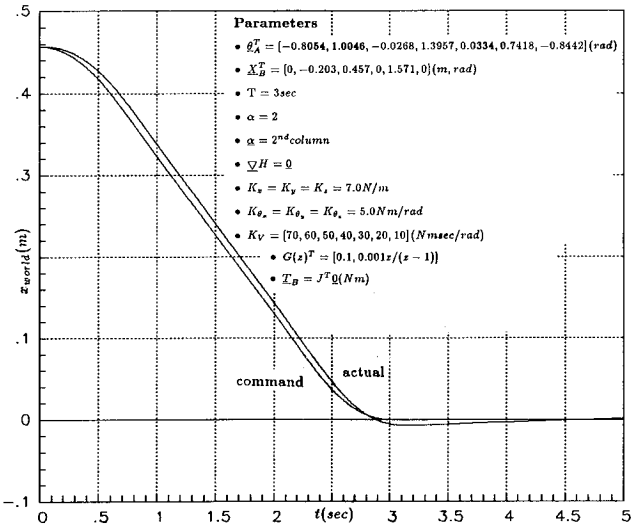



**Figure 5** Position ($x_{world}$) command/actual versus time – scenario number 1 (100 Hz, nominal $K/100$)

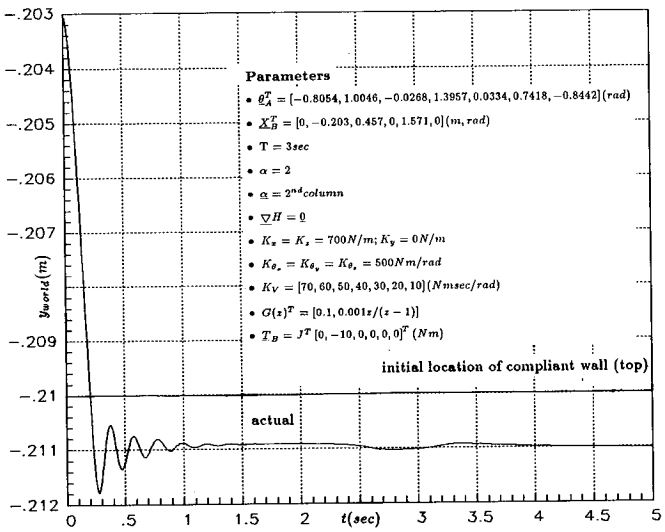



**Figure 6** Position ($y_{world}$) actual versus time – scenario number 2. (100 Hz, $K_{env} = 10kN/m$)

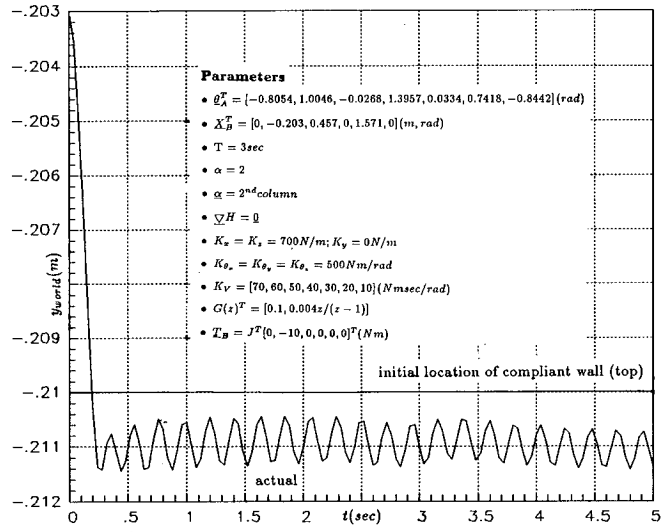



**Figure 7** Position ($y_{world}$) actual versus time – scenario number 2 (25Hz, $K_{env} = 10kN/m$)

(mentioned earlier), while all other blocks would be high speed custom blocks suitable for implementation. For the second issue, timing studies based on actual timing of the trajectory generator code (not provided here) indicate that 1 MC68020 $\mu$p could comfortably implement the entire algorithm at rates exceeding 100 Hz.

# 6 Conclusions and Future Work

The active stiffness controller when combined with the gradient projection technique for control of a particular 7 dof manipulator (Robotics Research T-710) appears to work satisfactorily based on preliminary simulation studies for noncontact situations and modest controller frequencies (e.g. $\geq 25Hz$). For the more difficult case where contact is established between the manipulator and the environment or for simple cooperative control, the system is stable when an individual manipulator's environment is somewhat compliant ($K_{env} \leq 1kN/m$) and the controller frequency is sufficiently high (i.e. $\geq 100Hz$). In addition, the computational intensity of the algorithm is quite low and timing studies suggest that 1 MC68020 $\mu$p could implement the algorithm at rates exceeding 100 Hz. Future work could entail: (1) using the torque compensation loop to achieve more stable results when performing contour following of fairly rigid surfaces (i.e. $K_{env} \geq 1kN/m$), (2) a more extensive local stability analysis, (3) include a more realistic actuator and contact dynamic model, and (4) verification on actual hardware. At the time of this writing some of this work is ongoing.

ORIGINAL PAGE IS
OF POOR QUALITY

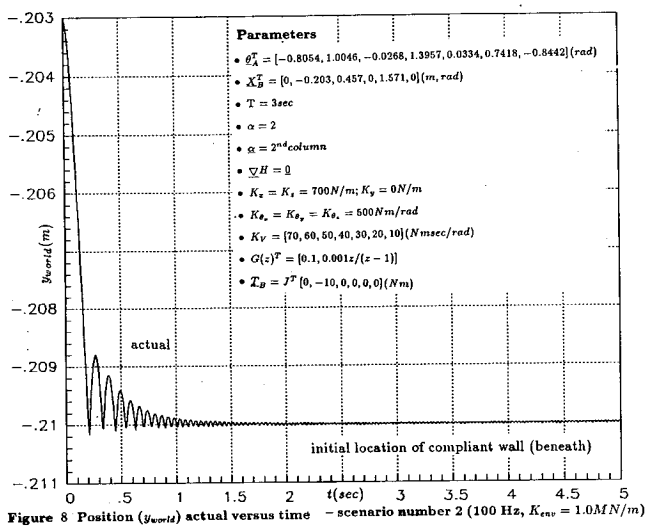



**Figure 8** Position ($y_{world}$) actual versus time – scenario number 2 (100 Hz, $K_{env} = 1.0MN/m$)

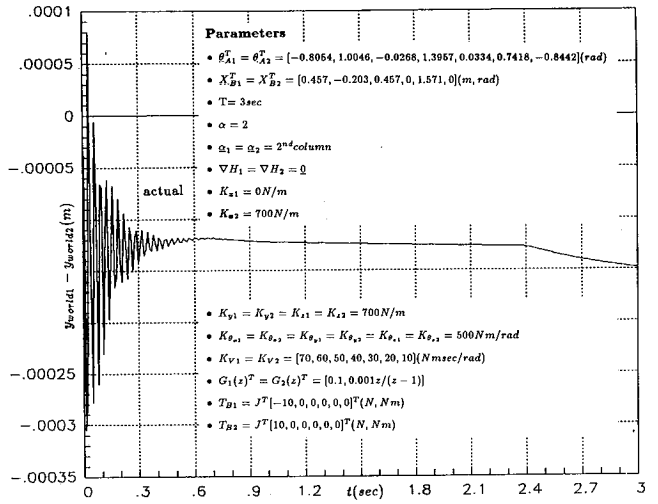



**Figure 9** Relative position error versus time - scenario number 3 ($100Hz$, $K_{spring} = 7,500N/m$)

# 7 Acknowledgements

The authors wish to acknowledge partial support of this work by the Defense Advanced Research Projects Agency (DARPA) under contract no. DAAA19-87-0094 and continued support by FMC Corporation under Independent Research and Development (IR and D) project no. 407. In addition, the authors would like to thank Professor O. Khatib of Stanford University for his consulting comments, Dan Whitney of Draper Laboratory for reviewing the paper, Paul Hasse for use of his graphical manipulator model, and Larry Coopet for performing the timing studies on the trajectory generator code.

# A Robotics Research T-710 Manipulator Kinematic Data (Representative)

- $\underline{h}_{world,base} = [0, 0, 0]^T(m)$
- $P^{Ref}_{world,base} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$
- $\underline{h}_{base,1} = [0, 0, 0]^T(m)$
- $P^{Ref}_{base,1} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
- $\underline{h}_{1,2} = [0.080, -0.099, 0]^T(m)$
- $P^{Ref}_{1,2} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
- $\underline{h}_{2,3} = [0.099, 0, 0.080]^T(m)$
- $P^{Ref}_{2,3} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}$
- $\underline{h}_{3,4} = [0.067, -0.231, 0]^T(m)$
- $P^{Ref}_{3,4} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
- $\underline{h}_{4,5} = [0.086, 0, 0.067]^T(m)$
- $P^{Ref}_{4,5} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}$
- $\underline{h}_{5,6} = [0, -0.244, 0]^T(m)$
- $P^{Ref}_{5,6} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
- $\underline{h}_{6,7} = [0, 0, 0]^T(m)$
- $P^{Ref}_{6,7} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}$
- $\underline{h}_{7,ee} = [-0.032, -0.292, 0]^T(m)$
- $P^{Ref}_{7,ee} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$



ORIGINAL PAGE IS OF POOR QUALITY

# B Robotics Research T-710 Manipulator Dynamic Data (Representative)

- link 1 mass $m_1 = 10$ kg
- link 1 center of gravity location $\underline{r}_{cg_{1,1}} = [0,0,0]^T$ (m)
- link 1 inertia tensor $I_{1,1} = \text{diag}[0.1,0.1,0.1]$ (kg$m^2$)
- link 2 mass $m_2 = 10$ kg
- link 2 center of gravity location $\underline{r}_{cg_{2,2}} = [0,0,0]^T$ (m)
- link 2 inertia tensor $I_{2,2} = \text{diag}[0.1,0.1,0.1]$ (kg$m^2$)
- link 3 mass $m_3 = 10$ kg
- link 3 center of gravity location $\underline{r}_{cg_{3,3}} = [0,0,0]^T$ (m)
- link 3 inertia tensor $I_{3,3} = \text{diag}[0.1,0.1,0.1]$ (kg$m^2$)
- link 4 mass $m_4 = 4$ kg
- link 4 center of gravity location $\underline{r}_{cg_{4,4}} = [0,0,0]^T$ (m)
- link 4 inertia tensor $I_{4,4} = \text{diag}[0.1,0.1,0.1]$ (kg$m^2$)
- link 5 mass $m_5 = 4$ kg
- link 5 center of gravity location $\underline{r}_{cg_{5,5}} = [0,0,0]^T$ (m)
- link 5 inertia tensor $I_{5,5} = \text{diag}[0.1,0.1,0.1]$ (kg$m^2$)
- link 6 mass $m_6 = 4$ kg
- link 6 center of gravity location $\underline{r}_{cg_{6,6}} = [0,0,0]^T$ (m)
- link 6 inertia tensor $I_{6,6} = \text{diag}[0.1,0.1,0.1]$ (kg$m^2$)
- link 7 mass $m_7 = 4$ kg
- link 7 center of gravity location $\underline{r}_{cg_{7,7}} = [0,0,0]^T$ (m)
- link 7 inertia tensor $I_{7,7} = \text{diag}[0.1,0.1,0.1]$ (kg$m^2$)
- end effector mass $m_{ee} = 1$ kg
- end effector center of gravity location $\underline{r}_{cg_{ee,ee}} = [0,0,0]^T$ (m)
- end effector inertia tensor $I_{ee,ee} = \text{diag}[0.1,0.1,0.1]$ (kg$m^2$)

# References


[1] Whitney, D.E., "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators," ASME Journal of Dynamic Systems, Measurement and Control, Dec. 1972, pp. 303-309.

[2] Baker, D.R. and Wampler, C. W., "Some Facts Concerning the Inverse Kinematics of Redundant Manipulators," IEEE International Conference on Robotics and Automation, 1987.

[3] Dubey, R., Euler, J., and Babcock, S., "An Efficient Gradient Projection Optimization Scheme for a Seven-Degree-of-Freedom Redundant Robot with Spherical Wrist," 1988 IEEE International Conference on Robotics and Automation, April, 1988.

[4] Zhang, Y. and Paul, R., "Robot Manipulator Control and Computational Cost," IEEE Workshop on Special Computer Architectures for Robotics, pp. 28-65, April 1988.

[5] An, C., Atkeson, C., and Hollarbach, J. 1988. *Model-Based Control of a Robot Manipulator.* Cambridge, Mass: MIT Press.

[6] Salisbury, J. K., "Active Stiffness Control of a Manipulator in Cartesian Coordinates," Proc. 19th IEEE Conference on Decision and Control, pp. 95-100, 1980.

[7] Karlen, J. Thompson, J. and Farrell, J., "Design and Control of Modular Kinematically-Redundant Manipulators," Second AIAA/NASA/USAF Symp. on Auto., Robotics and Adv. Comp. for the National Space Program, March 1987.

[8] Paul, R. 1981. *Robot Manipulators: Mathematics, Programming, and Control.* Cambridge, Mass: MIT Press.

[9] Integrated Systems Inc., "MATRIXx/AR (Automation and Robotics) Modeling, Control Design, and Simulation Program – User's Guide, May 1986.

[10] Hennessey, M. P., "Application of the Stiffness Controller and the Gradient Projection Technique to Control of a 7 DOF Manipulator," FMC Internal Report, Dec., 1988.

[11] Jackson, D., "Advanced Robotic Manipulator(s) Performance and Demonstration Test Plan – Draft," prepared for the U.S. ARMY AMCCOM by FMC Corporation, ASC, Minneapolis, MN, July 1988.


ORIGINAL PAGE IS OF POOR QUALITY